Ref #	Hits	Search Query	DBs	Default Operator	Plurals	Time Stamp
L1	2436	data adj (old or stale)	US-PGPUB; USPAT; EPO; JPO; IBM_TDB	OR	ON	2005/01/21 08:03
L2	625	data with (cache adj4 (old or stale or age))	US-PGPUB; USPAT; EPO; JPO; IBM_TDB	OR	ON	2005/01/21 08:04
L3	5	data with (cache adj4 ((old or stale or age)adj bit))	US-PGPUB; USPAT; EPO; JPO; IBM_TDB	OR	ON	2005/01/21 08:06
L4	1348	cache adj replac$5	US-PGPUB; USPAT; EPO; JPO; IBM_TDB	OR	ON	2005/01/21 08:06
L5	185	(cache adj replac$5) with (age or stale or old or timestamp or access)	US-PGPUB; USPAT; EPO; JPO; IBM_TDB	OR	ON	2005/01/21 08:07
L6	14	(cache adj replac$5) near (age or stale or old or timestamp or access)	US-PGPUB; USPAT; EPO; JPO; IBM_TDB	OR	ON	2005/01/21 08:15
L7	461	(cache adj replac$5) same (age or stale or old or timestamp or access)	US-PGPUB; USPAT; EPO; JPO; IBM_TDB	OR	ON	2005/01/21 08:07
L8	368	(number adj access) with cache	US-PGPUB; USPAT; EPO; JPO; IBM_TDB	OR	ON	2005/01/21 08:16
L9	596	(number adj access) same cache	US-PGPUB; USPAT; EPO; JPO; IBM_TDB	OR	ON	2005/01/21 08:16
L10	25	(number adj access) near cache	US-PGPUB; USPAT; EPO; JPO; IBM_TDB	OR	ON	2005/01/21 08:16
L11	0	((number adj access) near cache) with bit	US-PGPUB; USPAT; EPO; JPO; IBM_TDB	OR	ON	2005/01/21 08:17
L12	21	((number adj access) near cache) and bit	US-PGPUB; USPAT; EPO; JPO; IBM_TDB	OR	ON	2005/01/21 08:17

IEEE *Xplore*®
RELEASE 1.8

Welcome
United States Patent and Trademark Office

Your search matched **0** of **1117582** documents.
A maximum of **500** results are displayed, **15** to a page, sorted by **Relevance Descending** order.

Refine This Search:
You may refine your search by editing the current search expression or enteri
new one in the text box.

((number <near/1> access) <sentence> cache) <and> [Search

□ Check to search within this result set

Results Key:
JNL = Journal or Magazine **CNF** = Conference **STD** = Standard

Results:
No documents matched your query.

IEEE Xplore®
RELEASE 1.8

Welcome
United States Patent and Trademark Office

» Se

Welcome to IEEE Xplore®

- Home
- What Can I Access?
- Log-out

Tables of Contents

- Journals & Magazines
- Conference Proceedings
- Standards

Search

- By Author
- Basic
- Advanced
- CrossRef

Member Services

- Join IEEE
- Establish IEEE Web Account
- Access the IEEE Member Digital Library

IEEE Enterprise

- Access the IEEE Enterprise File Cabinet

Your search matched **0** of **1117582** documents.
A maximum of **500** results are displayed, **15** to a page, sorted by **Relevance Descending** order.

Refine This Search:
You may refine your search by editing the current search expression or enteri new one in the text box.

((number <near/2> access) <sentence> cache) <and> Search

□ Check to search within this result set

Results Key:
JNL = Journal or Magazine **CNF** = Conference **STD** = Standard

Results:
No documents matched your query.

IEEE HOME | SEARCH IEEE | SHOP | WEB ACCOUNT | CONTACT IEEE

IEEE

Membership Publications/Services Standards Conferences Careers/Jobs

IEEE Xplore® RELEASE 1.8

Welcome
United States Patent and Trademark Office

» Se

Help FAQ Terms IEEE Peer Review Quick Links

Welcome to IEEE Xplore®

- ○- Home
- ○- What Can I Access?
- ○- Log-out

Tables of Contents

- ○- Journals & Magazines
- ○- Conference Proceedings
- ○- Standards

Search

- ○- By Author
- ○- Basic
- ○- Advanced
- ○- CrossRef

Member Services

- ○- Join IEEE
- ○- Establish IEEE Web Account
- ○- Access the IEEE Member Digital Library

IEEE Enterprise

- ○- Access the IEEE Enterprise File Cabinet

Your search matched **2** of **1117582** documents.
A maximum of **500** results are displayed, **15** to a page, sorted by **Relevance Descending** order.

Refine This Search:
You may refine your search by editing the current search expression or enterii new one in the text box.

((number <sentence> access) <sentence> cache) <and Search

□ Check to search within this result set

Results Key:
JNL = Journal or Magazine **CNF** = Conference **STD** = Standard

1 **A low-power cache design for CalmRISC^{TM}-based systems**
Sangyeun Cho; Wooyoung Jung; Yongchun-Kim; Seh-Woong Jeong;
Computer Design, 2001. ICCD 2001. Proceedings. 2001 International Confere on , 23-26 Sept. 2001
Pages:394 - 399

[Abstract] [PDF Full-Text (656 KB)] **IEEE CNF**

2 **Software optimization of MPEG audio layer-III for a 32 bit RISC processor**
Wonchul Lee; Kisun You; Wonyong Sung;
Circuits and Systems, 2002. APCCAS '02. 2002 Asia-Pacific Conference on , Volume: 1 , 28-31 Oct. 2002
Pages:435 - 438 vol.1

[Abstract] [PDF Full-Text (648 KB)] **IEEE CNF**

Print Format

Home | Log-out | Journals | Conference Proceedings | Standards | Search by Author | Basic Search | Advanced Search | Join IEEE | Web Account | New this week | OPAC Linking Information | Your Feedback | Technical Support | Email Alerting | No Robots Please | Release Notes | IEEE Online Publications | Help | FAQ| Terms | Back to Top

PORTAL

US Patent & Trademark Office

Subscribe (Full Service) Register (Limited Service, Free) Login

Search: ● The ACM Digital Library ○ The Guide

+cache +line +replacement +bit SEARCH

THE ACM DIGITAL LIBRARY

Feedback Report a problem Satisfaction survey

Terms used **cache line replacement bit**

Found **1,075** of **148,786**

Sort results by: relevance

Display results: expanded form

Save results to a Binder

Search Tips

□ Open results in a new window

Try an Advanced Search

Try this search in The ACM Guide

Results 1 - 20 of 200

Result page: **1** 2 3 4 5 6 7 8 9 10 next

Best 200 shown

Relevance scale □ □ □ ■ ■

1 Session 3B: Compiler techniques in system level design: Software-assisted cache replacement mechanisms for embedded systems ■

Prabhat Jain, Srinivas Devadas, Daniel Engels, Larry Rudolph

November 2001 **Proceedings of the 2001 IEEE/ACM international conference on Computer-aided design**

Full text available: pdf(193.73 KB) Additional Information: full citation, abstract, references, citings, index terms

We address the problem of improving cache predictability and performance in embedded systems through the use of software-assisted replacement mechanisms. These mechanisms require additional software controlled state information that affects the cache replacement decision. Software instructions allow a program to kill a particular cache element, i.e., effectively make the element the least recently used element, or keep that cache element, i.e., the element will never be evicted.We prove basic th ...

2 Eliminating cache conflict misses through XOR-based placement functions ■

Antonio González, Mateo Valero, Nigel Topham, Joan M. Parcerisa

July 1997 **Proceedings of the 11th international conference on Supercomputing**

Full text available: pdf(1.21 MB) Additional Information: full citation, references, citings, index terms

Keywords: XOR-based placement functions, cache memory, conflict misses

3 Adapting cache line size to application behavior ■

Alexander V. Veidenbaum, Weiyu Tang, Rajesh Gupta, Alexandru Nicolau, Xiaomei Ji

May 1999 **Proceedings of the 13th international conference on Supercomputing**

Full text available: pdf(1.26 MB) Additional Information: full citation, references, citings, index terms

4 Cache-optimal methods for bit-reversals ■

Zhao Zhang, Xiaodong Zhang

January 1999 **Proceedings of the 1999 ACM/IEEE conference on Supercomputing (CDROM)**

Full text available: pdf(253.27 KB) Additional Information: full citation, references, index terms

5 A case for two-way skewed-associative caches
André Seznec
May 1993 **ACM SIGARCH Computer Architecture News , Proceedings of the 20th annual international symposium on Computer architecture**, Volume 21 Issue 2
Full text available: pdf(975.20 KB) Additional Information: full citation, references, citings, index terms

6 Capturing dynamic memory reference behavior with adaptive cache topology
Jih-Kwon Peir, Yongjoon Lee, Windsor W. Hsu
October 1998 **Proceedings of the eighth international conference on Architectural support for programming languages and operating systems**, Volume 33 , 32 Issue 11 , 5
Full text available: pdf(1.50 MB) Additional Information: full citation, abstract, references, citings, index terms

Memory references exhibit locality and are therefore not uniformly distributed across the sets of a cache. This skew reduces the effectiveness of a cache because it results in the caching of a considerable number of less-recently-used lines which are less likely to be re-referenced before they are replaced. In this paper, we describe a technique that dynamically identifies these less-recently-used lines and effectively utilizes the cache frames they occupy to more accurately approximate the glob ...

7 LRU-based column-associative caches
Byung-Kwon Chung, Jih-Kwon Peir
May 1998 **ACM SIGARCH Computer Architecture News**, Volume 26 Issue 2
Full text available: pdf(859.94 KB) Additional Information: full citation, abstract, citings, index terms

The column-associative cache is a direct-mapped cache that may be accessed more than once, each time with a different hash function, to satisfy a memory request. In the column-associative cache, the possible locations that a line can reside in defines the column. The original scheme relies on a rehash bit array to guide the replacement policy within the column. A subsequent proposal uses an index vector to speed up the cache search within the column. In this paper, we consider the idea of using ...

8 Cache replacement with dynamic exclusion
Scott McFarling
April 1992 **ACM SIGARCH Computer Architecture News , Proceedings of the 19th annual international symposium on Computer architecture**, Volume 20 Issue 2
Full text available: pdf(894.10 KB) Additional Information: full citation, abstract, references, citings, index terms

Most recent cache designs use direct-mapped caches to provide the fast access time required by modern high speed CPU's. Unfortunately, direct-mapped caches have higher miss rates than set-associative caches, largely because direct-mapped caches are more sensitive to conflicts between items needed frequently in the same phase of program execution. This paper presents a new technique for reducing direct-mapped cache misses caused by conflicts for a particular cache line. A small fi ...

9 Set-associative cache simulation using generalized binomial trees
Rabin A. Sugumar, Santosh G. Abraham
February 1995 **ACM Transactions on Computer Systems (TOCS)**, Volume 13 Issue 1
Full text available: pdf(1.51 MB) Additional Information: full citation, abstract, references, citings, index terms, review

Set-associative caches are widely used in CPU memory hierarchies, I/O subsystems, and file systems to reduce average access times. This article proposes an efficient simulation technique for simulating a group of set-associative caches in a single pass through the address trace, where all caches have the same line size but varying associativities and varying number of sets. The article also introduces a generalization of the ordinary binomial tree and presents a representation of caches in ...

Keywords: all-associativity simulation, binomial tree, cache modeling, inclusion properties, set-associative caches, single-pass simulation, trace-driven simulation

10 CAT—caching address tags: a technique for reducing area cost of on-chip caches
Hong Wang, Tong Sun, Qing Yang
May 1995 **ACM SIGARCH Computer Architecture News , Proceedings of the 22nd annual international symposium on Computer architecture**, Volume 23 Issue 2
Full text available: pdf(1.36 MB) Additional Information: full citation, abstract, references, citings, index terms

This paper presents a technique for minimizing chip-area cost of implementing an on-chip cache memory of microprocessors. The main idea of the technique *Caching Address Tags, or CAT cache* for short. The *CAT* cache exploits locality property that exists among addresses of memory references for the purpose of minimizing chip area-cost of address tags. By keeping only a limited number of distinct tags of cached data rather than having as many tags as cache lines, the *CAT* ...

11 A new cache replacement scheme based on backpropagation neural networks
Humayun Khalid
March 1997 **ACM SIGARCH Computer Architecture News**, Volume 25 Issue 1
Full text available: pdf(572.97 KB) Additional Information: full citation, abstract, index terms

In this paper, we present a new neural network-based algorithm, **KORA** (*K*halid Shad*O*w *R*eplacement *A*lgorithm), that uses backpropagation neural network (BPNN) for the purpose of guiding the line/block replacement decisions in cache. This work is a continuation of our previous research presented in [1]-[3]. The KORA algorithm attempts to approximate the replacement decisions made by the optimal scheme (OPT). The key to our algorithm is to identify and subsequently ...

Keywords: cache memory, neural networks, performance evaluation

12 Performance of the KORA-2 cache replacement scheme
Humayun Khalid
September 1997 **ACM SIGARCH Computer Architecture News**, Volume 25 Issue 4
Full text available: pdf(630.87 KB) Additional Information: full citation, abstract, index terms

In this paper, we propose a new strategy (KORA-2) for the replacement of lines in cache memories. The algorithm is efficient and easily implementable. It is basically an extension of our previous work presented in [1]-[5]. Key to our algorithm is to identify and discard inactive lines relatively quickly as opposed to the conventional replacement algorithms. Trace-driven simulations were performed for 42 different cache configurations using benchmark programs from SPEC92 (Standard performance Eva ...

Keywords: cache memory, neural networks, performance evaluation

13
Application-specific memory management for embedded systems using software-

controlled caches
Derek Chiou, Prabhat Jain, Larry Rudolph, Srinivas Devadas
June 2000 **Proceedings of the 37th conference on Design automation**

Full text available: pdf(76.30 KB) Additional Information: full citation, abstract, references, citings, index terms

We propose a way to improve the performance of embedded processors running data-intensive applications by allowing software to allocate on-chip memory on an application-specific basis. On-chip memory in the form of cache can be made to act like scratch-pad memory via a novel hardware mechanism, which we call column caching. Column caching enables dynamic cache partitioning in software, by mapping data regions to a specified sets of cache "columns" or "ways ...

14 An integrated memory management scheme for dynamic alias resolution
Tzi-cker Chiueh
August 1991 **Proceedings of the 1991 ACM/IEEE conference on Supercomputing**

Full text available: pdf(944.90 KB) Additional Information: full citation, references, citings, index terms

15 Next cache line and set prediction
Brad Calder, Dirk Grunwald
May 1995 **ACM SIGARCH Computer Architecture News , Proceedings of the 22nd annual international symposium on Computer architecture**, Volume 23 Issue 2

Full text available: pdf(1.25 MB) Additional Information: full citation, abstract, references, citings, index terms

Accurate instruction fetch and branch prediction is increasingly important on today's wide-issue architectures. Fetch prediction is the process of determining the next instruction to request from the memory subsystem. Branch prediction is the process of predicting the likely out-come of branch instructions. Several researchers have proposed very effective fetch and branch prediction mechanisms including branch target buffers (BTB) that store the target addresses of taken branches. An alternative ...

16 Efficient simulation of caches under optimal replacement with applications to miss characterization
Rabin A. Sugumar, Santosh G. Abraham
June 1993 **ACM SIGMETRICS Performance Evaluation Review , Proceedings of the 1993 ACM SIGMETRICS conference on Measurement and modeling of computer systems**, Volume 21 Issue 1

Full text available: pdf(1.26 MB) Additional Information: full citation, references, citings, index terms

17 Architecture & distributed systems: Performance evaluation of cache replacement policies for the SPEC CPU2000 benchmark suite
Hussein Al-Zoubi, Aleksandar Milenkovic, Milena Milenkovic
April 2004 **Proceedings of the 42nd annual Southeast regional conference**

Full text available: pdf(251.03 KB) Additional Information: full citation, abstract, references

Replacement policy, one of the key factors determining the effectiveness of a cache, becomes even more important with latest technological trends toward highly associative caches. The state-of-the-art processors employ various policies such as Random, Least Recently Used (LRU), Round-Robin, and PLRU (Pseudo LRU), indicating that there is no common wisdom about the best one. Optimal yet unattainable policy would replace cache memory block whose next reference is the farthest away in the future, a ...

Keywords: cache memory, performance evaluation, replacement policy

18 Runtime identification of cache conflict misses: The adaptive miss buffer

Jamison D. Collins, Dean M. Tullsen

November 2001 **ACM Transactions on Computer Systems (TOCS)**, Volume 19 Issue 4

Full text available: pdf(1.08 MB) Additional Information: full citation, abstract, references, index terms

This paper describes the miss classification table, a simple mechanism that enables the processor or memory controller to identify each cache miss as either a conflict miss or a capacity (non-conflict) miss. The miss classification table works by storing part of the tag of the most recently evicted line of a cache set. If the next miss to that cache set has a matching tag, it is identified as a conflict miss. This technique correctly identifies 88&percnt; of misses.Several applications of this i ...

Keywords: Cache architecture, adaptive miss buffer, cache exclusion, conflict misses, prefetching, victim cache

19 The difference-bit cache

Toni Juan, Tomás Lang, Juan J. Navarro

May 1996 **ACM SIGARCH Computer Architecture News , Proceedings of the 23rd annual international symposium on Computer architecture**, Volume 24 Issue 2

Full text available: pdf(798.80 KB) Additional Information: full citation, abstract, references, citings, index terms

The difference-bit cache is a two-way set-associative cache with an access time that is smaller than that of a conventional one and close or equal to that of a direct-mapped cache. This is achieved by noticing that the two tags for a set have to differ at least by one bit and by using this bit to select the way. In contrast with previous approaches that predict the way and have two types of hits (primary of one cycle and secondary of two to four cycles), all hits of the difference-bit cache are ...

20 Using virtual lines to enhance locality exploitation

O. Temam, Y. Jegou

July 1994 **Proceedings of the 8th international conference on Supercomputing**

Full text available: pdf(1.15 MB) Additional Information: full citation, abstract, references, citings, index terms

Because the spatial locality of numerical codes is significant, the potential for performance improvements is important. However, large cache lines cannot be used in current on-chip data caches because of the important pollution they breed. In this paper, we propose a hardware design, called the Virtual Line Scheme, that allows the utilization of large virtual cache lines when fetching data from memory for better exploitation of spatial locality, while the ...

Keywords: cache architecture, memory hierarchy, numerical codes, spatial locality, temporal locality

Results 1 - 20 of 200 Result page: **1** 2 3 4 5 6 7 8 9 10 next

Useful downloads: Adobe Acrobat QuickTime Windows Media Player Real Player

Terms used **cache line replacement bit** and **number** and **access** Found **1,075** of **148,786**

Sort results by: relevance
Display results: expanded form

Save results to a Binder
Search Tips
□ Open results in a new window

Try an Advanced Search
Try this search in The ACM Guide

Results 1 - 20 of 200 Result page: **1** 2 3 4 5 6 7 8 9 10 next

Best 200 shown Relevance scale □□■■■

1 Session 3B: Compiler techniques in system level design: Software-assisted cache replacement mechanisms for embedded systems ■
Prabhat Jain, Srinivas Devadas, Daniel Engels, Larry Rudolph
November 2001 **Proceedings of the 2001 IEEE/ACM international conference on Computer-aided design**
Full text available: pdf(193.73 KB) Additional Information: full citation, abstract, references, citings, index terms

We address the problem of improving cache predictability and performance in embedded systems through the use of software-assisted replacement mechanisms. These mechanisms require additional software controlled state information that affects the cache replacement decision. Software instructions allow a program to kill a particular cache element, i.e., effectively make the element the least recently used element, or keep that cache element, i.e., the element will never be evicted.We prove basic th ...

2 Reducing power in superscalar processor caches using subbanking, multiple line buffers and bit-line segmentation ■
Kanad Ghose, Milind B. Kamble
August 1999 **Proceedings of the 1999 international symposium on Low power electronics and design**
Full text available: pdf(789.43 KB) Additional Information: full citation, references, citings, index terms

Keywords: low power caches, power estimation

3 Cache Memories ■
Alan Jay Smith
September 1982 **ACM Computing Surveys (CSUR)**, Volume 14 Issue 3
Full text available: pdf(4.61 MB) Additional Information: full citation, references, citings, index terms

4 Eliminating cache conflict misses through XOR-based placement functions ■
Antonio González, Mateo Valero, Nigel Topham, Joan M. Parcerisa
July 1997 **Proceedings of the 11th international conference on Supercomputing**

Full text available: pdf(1.21 MB) Additional Information: full citation, references, citings, index terms

Keywords: XOR-based placement functions, cache memory, conflict misses

5 Capturing dynamic memory reference behavior with adaptive cache topology
Jih-Kwon Peir, Yongjoon Lee, Windsor W. Hsu
October 1998 **Proceedings of the eighth international conference on Architectural support for programming languages and operating systems**, Volume 33 , 32 Issue 11 , 5

Full text available: pdf(1.50 MB) Additional Information: full citation, abstract, references, citings, index terms

Memory·references exhibit locality and are therefore not uniformly distributed across the sets of a cache. This skew reduces the effectiveness of a cache because it results in the caching of a considerable number of less-recently-used lines which are less likely to be re-referenced before they are replaced. In this paper, we describe a technique that dynamically identifies these less-recently-used lines and effectively utilizes the cache frames they occupy to more accurately approximate the glob ...

6 Architecture & distributed systems: Performance evaluation of cache replacement policies for the SPEC CPU2000 benchmark suite
Hussein Al-Zoubi, Aleksandar Milenkovic, Milena Milenkovic
April 2004 **Proceedings of the 42nd annual Southeast regional conference**

Full text available: pdf(251.03 KB) Additional Information: full citation, abstract, references

Replacement policy, one of the key factors determining the effectiveness of a cache, becomes even more important with latest technological trends toward highly associative caches. The state-of-the-art processors employ various policies such as Random, Least Recently Used (LRU), Round-Robin, and PLRU (Pseudo LRU), indicating that there is no common wisdom about the best one. Optimal yet unattainable policy would replace cache memory block whose next reference is the farthest away in the future, a ...

Keywords: cache memory, performance evaluation, replacement policy

7 Cache replacement with dynamic exclusion
Scott McFarling
April 1992 **ACM SIGARCH Computer Architecture News , Proceedings of the 19th annual international symposium on Computer architecture**, Volume 20 Issue 2

Full text available: pdf(894.10 KB) Additional Information: full citation, abstract, references, citings, index terms

Most recent cache designs use direct-mapped caches to provide the fast access time required by modern high speed CPU's. Unfortunately, direct-mapped caches have higher miss rates than set-associative caches, largely because direct-mapped caches are more sensitive to conflicts between items needed frequently in the same phase of program execution. This paper presents a new technique for reducing direct-mapped cache misses caused by conflicts for a particular cache line. A small fi ...

8 HAC: hybrid adaptive caching for distributed storage systems
Miguel Castro, Atul Adya, Barbara Liskov, Andrew C. Meyers
October 1997 **ACM SIGOPS Operating Systems Review , Proceedings of the sixteenth ACM symposium on Operating systems principles**, Volume 31 Issue 5

Full text available: pdf(2.21 MB) Additional Information: full citation, references, citings, index terms

9 Next cache line and set prediction

Brad Calder, Dirk Grunwald

May 1995 **ACM SIGARCH Computer Architecture News , Proceedings of the 22nd annual international symposium on Computer architecture**, Volume 23 Issue 2

Full text available: pdf(1.25 MB) Additional Information: full citation, abstract, references, citings, index terms

Accurate instruction fetch and branch prediction is increasingly important on today's wide-issue architectures. Fetch prediction is the process of determining the next instruction to request from the memory subsystem. Branch prediction is the process of predicting the likely out-come of branch instructions. Several researchers have proposed very effective fetch and branch prediction mechanisms including branch target buffers (BTB) that store the target addresses of taken branches. An alternative ...

10 Towards a theory of cache-efficient algorithms

Sandeep Sen, Siddhartha Chatterjee, Neeraj Dumir

November 2002 **Journal of the ACM (JACM)**, Volume 49 Issue 6

Full text available: pdf(273.41 KB) Additional Information: full citation, abstract, references, index terms

We present a model that enables us to analyze the running time of an algorithm on a computer with a memory hierarchy with limited associativity, in terms of various cache parameters. Our cache model, an extension of Aggarwal and Vitter's I/O model, enables us to establish useful relationships between the cache complexity and the I/O complexity of computations. As a corollary, we obtain cache-efficient algorithms in the single-level cache model for fundamental problems like sorting, FFT, and an i ...

Keywords: Hierarchical memory, I/O complexity, lower bound

11 Area efficient architectures for information integrity in cache memories

Seongwoo Kim, Arun K. Somani

May 1999 **ACM SIGARCH Computer Architecture News , Proceedings of the 26th annual international symposium on Computer architecture**, Volume 27 Issue 2

Full text available: pdf(227.09 KB) Publisher Site Additional Information: full citation, abstract, references, citings, index terms

Information integrity in cache memories is a fundamental requirement for dependable computing. Conventional architectures for enhancing cache reliability using check codes make it difficult to trade between the level of data integrity and the chip area requirement. We focus on transient fault tolerance in primary cache memories and develop new architectural solutions, to maximize fault coverage when the budgeted silicon area is not sufficient for the conventional configuration of an error checki ...

12 The difference-bit cache

Toni Juan, Tomás Lang, Juan J. Navarro

May 1996 **ACM SIGARCH Computer Architecture News , Proceedings of the 23rd annual international symposium on Computer architecture**, Volume 24 Issue 2

Full text available: pdf(798.80 KB) Additional Information: full citation, abstract, references, citings, index terms

The difference-bit cache is a two-way set-associative cache with an access time that is smaller than that of a conventional one and close or equal to that of a direct-mapped cache. This is achieved by noticing that the two tags for a set have to differ at least by one bit and by using this bit to select the way. In contrast with previous approaches that predict the way and have two types of hits (primary of one cycle and secondary of two to four cycles), all hits of the difference-bit cache are ...

13 A novel cache design for vector processing
Qing Yang, Liping Wu Yang
April 1992 **ACM SIGARCH Computer Architecture News , Proceedings of the 19th annual international symposium on Computer architecture**, Volume 20 Issue 2
Full text available: pdf(1.35 MB) Additional Information: full citation, abstract, references, citings, index terms

This paper introduces an innovative cache design for vector computers, called prime-mapped cache. By utilizing the special properties of a Mersenne prime, the new design does not increase the critical path length of a processor, nor does it increase the cache access time as compared to a direct-mapped cache. The prime-mapped cache minimizes cache miss ratio caused by line interferences that have been shown to be critical for numerical applications by previous investigators. We show that sig ...

14 CAT—caching address tags: a technique for reducing area cost of on-chip caches
Hong Wang, Tong Sun, Qing Yang
May 1995 **ACM SIGARCH Computer Architecture News , Proceedings of the 22nd annual international symposium on Computer architecture**, Volume 23 Issue 2
Full text available: pdf(1.36 MB) Additional Information: full citation, abstract, references, citings, index terms

This paper presents a technique for minimizing chip-area cost of implementing an on-chip cache memory of microprocessors. The main idea of the technique *Caching Address Tags, or CAT cache* for short. The *CAT* cache exploits locality property that exists among addresses of memory references for the purpose of minimizing chip area-cost of address tags. By keeping only a limited number of distinct tags of cached data rather than having as many tags as cache lines, the *CAT* ...

15 A bit-slice cache controller
B. D. Ackland
April 1979 **Proceedings of the 6th annual symposium on Computer architecture**
Full text available: pdf(549.11 KB) Additional Information: full citation, abstract, references, citings, index terms

Cache storage is a proven memory speedup technique in large mainframe computers. Two of the main difficulties associated with the use of this concept in small machines are the high relative cost and complexity of the cache controller. An LSI bit-slice chip set is described which should reduce both controller cost and complexity. The set will enable a memory designer to construct a wide variety of cache structures with a minimum number of components and interconnections. Design parameters ar ...

16 Cache-optimal methods for bit-reversals
Zhao Zhang, Xiaodong Zhang
January 1999 **Proceedings of the 1999 ACM/IEEE conference on Supercomputing (CDROM)**
Full text available: pdf(253.27 KB) Additional Information: full citation, references, index terms

17 The pool of subsectors cache design
Jeffrey B. Rothman, Alan Jay Smith
May 1999 **Proceedings of the 13th international conference on Supercomputing**
Full text available: pdf(1.69 MB) Additional Information: full citation, references, citings, index terms

18 A new page table for 64-bit address spaces
M. Talluri, M. D. Hill, Y. A. Khalidi
December 1995 **ACM SIGOPS Operating Systems Review , Proceedings of the fifteenth ACM symposium on Operating systems principles**, Volume 29 Issue 5
Full text available: pdf(1.97 MB) Additional Information: full citation, references, citings, index terms

19 Runtime identification of cache conflict misses: The adaptive miss buffer
Jamison D. Collins, Dean M. Tullsen
November 2001 **ACM Transactions on Computer Systems (TOCS)**, Volume 19 Issue 4
Full text available: pdf(1.08 MB) Additional Information: full citation, abstract, references, index terms

This paper describes the miss classification table, a simple mechanism that enables the processor or memory controller to identify each cache miss as either a conflict miss or a capacity (non-conflict) miss. The miss classification table works by storing part of the tag of the most recently evicted line of a cache set. If the next miss to that cache set has a matching tag, it is identified as a conflict miss. This technique correctly identifies 88&percnt; of misses.Several applications of this i ...

Keywords: Cache architecture, adaptive miss buffer, cache exclusion, conflict misses, prefetching, victim cache

20 Energy efficient memory systems: Energy efficient frequent value data cache design
Jun Yang, Rajiv Gupta
November 2002 **Proceedings of the 35th annual ACM/IEEE international symposium on Microarchitecture**
Full text available: pdf(1.08 MB) Publisher Site Additional Information: full citation, abstract, references, citings, index terms

Recent work has shown that a small number of distinct frequently occurring values often account for a large portion of memory accesses. In this paper we demonstrate how this frequent value phenomenon can be exploited in designing a cache that trades off performance with energy efficiency. We propose the design of the Frequent Value Cache (FVC) in which storing a frequent value requires few bits as they are stored in encoded form while all other values are stored in unencoded form using 32 bits. ...

Results 1 - 20 of 200 Result page: **1** 2 3 4 5 6 7 8 9 10 next